

REMARKS

The application includes claims 1-24 prior to entering this amendment. Claims 1-24 remain in the case for reconsideration. No new subject matter has been added.

Claim Objections

The examiner objects to claims 1 and 9 for informalities. The words "A method" have been changed to "The method" in claim 9 and the words "that initiate a subgroup" have been corrected to "that initiate a subgroup" to show the proper changes from the previous office action response.

Claim Rejections - 35 U.S.C. § 102

The examiner rejects claims 1-2, 4-6 and 8-24 under 35 U.S.C. § 102(b) as being anticipated by Chang, *et al.*, (U.S. Patent 5,745,747).

Claim 1 has been amended and now recites:

a processor configured to:
associate multiple different activities with a same transaction, each of the multiple different activities each consisting of a separate different associated subgroup of program instructions for the same transaction,
for each different subgroup of program instructions, initiate a different associated subgroup of multiple different read and/or write actions ~~on~~ that access an associated group of multiple different data items;
use and assign only one single separate lock duration for all of the multiple different data items associated with each different subgroup of program instructions associated with each of the different activities;
maintain the multiple different locks on all of the multiple different data items associated with the same activities and then releasing all of the multiple different locks for all of the different data items associated with the same activities together only when all of the subgroup of program instructions associated with the same activities are completed so that all of the multiple different locks on all of the multiple different data items associated with the same activities have a same lock duration; and
release all of the locks on a first set of multiple different data items associated with a first activity of the transaction while a second set of data items that include at least some of the first set of data items from the first activity, but that are associated with a second activity for the same transaction, remain locked for a second separate single lock duration associated with a second activity.

This is clearly shown in FIGS. 2 and 4-7 where specifically in FIG. 2 the same activity 14 is associated with multiple different read and/or write operations 15 that access multiple different data items 16. Also refer to page 6, lines 1-10 where it is described that different activities 14

may overlap in time duration, that a transaction can have arbitrary many concurrent activities 14, and that any activity 14 may lock an arbitrary number of data items 16.

Chang describes a data processing program that supports multiple processes per transaction (col. 3, lines 8-9). Each transaction can have multiple processes and each process can request a resource lock (col. 3, lines 19-22).

Processes send lock requests to a lock manager (col. 3, line 27). The lock manager maintains a separate queue of lock requests for each lockable resource (col. 3, lines 28-29). When the lock manager receives a lock request, it allocates a lock request block (LRB) for that particular requested resource and stores the LRB in a queue associated with the particular requested resource (col. 3, lines 30-35).

A granted transaction mode (GTM) field in the LRBs store an upper bound of the lock mode granted to the transaction owning the LRB. The lock manager compares the different fields in the different LRBs assigned to the same resource, to determine if requested lock modes are compatible with the GTM assigned to the data field (col. 3, lines 53- col. 4, line 19). If there are multiple LRBs belonging to the same transaction as a current LRB, the lock manager stores the upper bound for all of the process lock requests in the first LRB of that transaction in the queue (col. 4, lines 27-31). When a transaction commits or rolls back, the lock manager uses the transaction ID to release all locks belonging to the transaction (col. 4, lines 39-41).

The lock duration associated with a transaction commit or roll back as described in Chang is only associated with a single data field (col. 4, lines 39-40). This requires the lock manager in Chang to maintain separate lock queues for each separate data field accessed by a process or transaction. There is no suggestion in Chang of locks assigned to multiple different data items all associated with a same activity of related read and/or write accesses that all have

common lock duration as recited in claim 1. Conversely, the granted transaction mode (GTM) for each individual data field in Chang stores the upper bound of the lock mode granted to the transaction owning the LRB (col. 3, lines 49-50).

The complexity of the lock manager system required in Chang is exactly one of the problems overcome in the present application. Refer to the present application at page 3, lines 7-12 describing how cumbersome and complicated it is for a queue manager to have to note and manage all classifications for data items having a priori lock durations. Also, refer to the specification at page 3, line 7 stating that currently there is no meaningful way to define and manage a potentially unlimited number of groups which a transaction may want to classify a given lock.

As described above, the lock manager in Chang creates LRBs for each individual lock request for each individual data field. The LRBs each have an associated lock duration that has to be compared with all of the other lock requests for the same resource (col. 3, lines 37-43). The present application overcomes this problem by associating arbitrary groups of multiple different data items all associated with a common activity with same single lock duration.

Because Chang uses individual lock request queues for each data field (col. 3, lines 27-30), there simply is no feasible way to associate multiple different resources of Chang with the same lock duration. Hypothetically, if Chang were to provide single lock duration for multiple different resources (which Chang does not), then every single data field (resource) for the entire database system would have to be checked and LRB fields recalculated each time there was a new lock request for any resource. This would simply be unmanageable.

There is no suggestion in Chang of setting an arbitrary lock duration associated with all of the different data items in the same related activity as recited in claim 1. For example,

nowhere does Chang even suggest comparing the process ID and transaction ID for different resources.

Refer to the present application on page 8, lines 1-22 where execute commands can arbitrarily associate all of the data items accessed by two different activities in the same transaction with multiple different data items. Also refer to FIGS. 5 and 6 where buffers 60 are used for identifying the multiple different data items that are all associated with the same activities and transactions. None of this is suggested in Chang.

Regardless, there is also no suggestion in Chang of releasing locks for multiple different data items associated with a same activity while *a second set of data items that include at least some of the first set of data items from the first activity, but that are associated with a second activity for the same transaction, remain locked for a second separate single lock duration associated with a second activity* as also recited in claim 1.

Even if the LRBs for multiple different data fields in Chang were somehow associated together (which they're not), all of the locks for all of the data items in the same transaction are all be released at one time when the transaction commits or rolls back (col. 4, lines 38-41). Conversely, different activities as recited in claim 1 of the present application can overlap such that lock durations for data items in non-overlapping activities can be released prior to completion of the entire transaction. See FIG. 5 of the present application.

Thus, as recited in claim 1, data items from a first activity that are not accessed by the second activity are not needlessly locked for the duration of the entire transaction as they would be in Chang.

Accordingly, claim 1 is patentable under 35 U.S.C. § 102(b) over Chang. The other independent claims include at least some of the same elements as claim 1 and are therefore patentable for at least some of the same reasons.

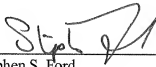
Conclusion

For the foregoing reasons, reconsideration and allowance of claims 1-24 of the application as amended is requested. The examiner is encouraged to telephone the undersigned at (503) 224-2170 if it appears that an interview would be helpful in advancing the case.

Customer No. 73552

Respectfully submitted,

STOLOWITZ FORD COWGER LLP



Stephen S. Ford
Reg. No. 35,319

STOLOWITZ FORD COWGER LLP
621 SW Morrison Street, Suite 600
Portland, OR 97205
(503) 224-2170